This correspondence is being deposited with the United States Postal Service as Express Mail addressed to: Box Patent Application, Assistant Commissioner for Patents; Washington, D. C. 20231, on March 9, 2001, Express Mail receipt No. EF055069983US.

REDUCING DELAYS ASSOCIATED WITH INSERTING A CHECKSUM INTO A NETWORK MESSAGE

Stephen E. J. Blightman

Laurence B. Boucher

Peter K. Craft

David A. Higgen

Clive M. Philbrick

Daryl D. Starr

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit under 35 U.S.C. §120 of U.S. Patent Application Serial No. 09/464,283, filed December 15, 1999, by Laurence B. Boucher et al., which in turn claims the benefit under 35 U.S.C. §120 of U.S. Patent Application Serial No. 09/439,603, filed November 12, 1999, by Laurence B. Boucher et al., which in turn claims the benefit under 35 U.S.C. § 119(e)(1) of the Provisional Application Serial No. 60/061,809, filed on October 14, 1997. This application also claims the benefit under 35 U.S.C. §120 of U.S. Patent Application Serial No. 09/384,792, filed August 27, 1999, which in turn claims the benefit under 35 U.S.C. § 119(e)(1) of the Provisional Application Serial No. 60/098,296, filed August 27, 1998. This application also claims the benefit under 35 U.S.C. §120 of U.S. Patent Application Serial No. 09/067,544, filed April 27, 1998. The subject matter of all the above-identified patent applications, and of the two above-identified provisional applications, is incorporated by reference herein.

BACKGROUND INFORMATION

[0003] Figure 1 (Prior Art) is a simplified diagram of a TCP packet. Figure 2 is a simplified diagram of a network interface card (NIC) 100 card called an intelligent network interface card (INIC). One of the operations the INIC performs is to read data for a TCP packet out of host memory 101 on a host computer 102 and to transmit that data as the data payload of a TCP message onto a network 103.

[0004] A difficulty associated with performing this operation quickly is that the checksum of the TCP packet is located near the front of the packet before the data payload. The checksum is a function of all the data of the data payload. Consequently all the data of the payload must generally be transferred to the INIC 100 before the checksum can be generated. Consequently, in general, all the data of the payload is received onto the INIC card, the checksum 104 is generated, the checksum 104 is then combined with the data payload in DRAM 105 to form the complete TCP packet 106, and the complete TCP packet 106 is then transferred from DRAM 105 to the network 107.

[0005] Figure 2 illustrates this flow of information. Arrow 108 illustrates the flow of data from host memory 101 and across PCI bus 103 to DRAM 105 located on INIC card 100. While the data is being transferred, processor 109 on INIC card 100 builds the TCP header 110 in faster SRAM 111. The TCP header is formed in SRAM 111 rather than DRAM 105 because processor 109 needs to perform multiple operations on the header 110 as it is assembled and doing such multiple operations from relatively slow DRAM would unduly slow down processor 109. When all the data has been received onto the INIC 100, processor 109 is able to determine the checksum 104. The complete TCP header 110

including the correct checksum 104 is at that point residing in SRAM 111. Arrow 112 represents the assembly and writing of the complete header 110 from processor 109 to SRAM 111.

[0006] Once the complete header 110 is assembled, it is transferred from SRAM 111 to DRAM 105 in a relatively slow write to DRAM 105. Arrow 113 illustrated this transfer. Once the complete TCP packet 106 is assembled in DRAM 105, the complete packet 106 is output from DRAM 105 to the network 107. In the example of Figure 2, this transfer is represented by arrow 114.

[0007] Unfortunately, the writing to DRAM 105 is often a relatively slow process and this writing can only begin once all the data has been received onto the INIC card. The result is an undesirable latency in the outputting of the TCP packet onto the network. A solution is desired.

SUMMARY

[0008] A first partial checksum for the header portion of a TCP header is generated on an intelligent network interface card (INIC) before all the data of the data payload of the TCP message has been transferred to the INIC. A pseudopacket with the first partial checksum and the data is assembled in DRAM on the INIC as the data arrives onto the INIC. When the last portion of the data of the data payload is received onto the INIC, a second partial checksum for the data payload is generated. This second partial checksum is not, however, written into DRAM. Rather, the pseudopacket is read out of DRAM for transfer to the network and while the pseudopacket is being transferred the second partial header is combined with the first partial header such that the resulting final TCP

checksum is inserted into the pseudopacket. The pseudopacket is therefore converted into a complete TCP packet with a correct checksum as it is output from the INIC to the network.

[0009] In this way, the slow write to DRAM of the complete TCP header after the payload has already been transferred to DRAM is avoided. Rather than generating the complete TCP checksum and taking the time to write it into DRAM, the complete TCP checksum is generated on the fly as the pseudopacket is transferred from DRAM to the network.

[0010] This summary does not purport to define the invention. The claims, and not this summary, define the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Figure 1 is a simplified diagram of a TCP packet.

[0012] Figure 2 is a diagram used in explaining the background of the invention.

[0013] Figure 3 is a diagram of an intelligent network interface card (INIC) in accordance with an embodiment of the present invention.

[0014] Figure 4 is a diagram of a method in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0015] Figure 3 is a diagram of an intelligent network interface card (INIC) 200 in accordance with one embodiment of the present invention. INIC 200 is coupled to host computer 201 via PCI bus 202. For additional information on INIC 200, see U.S. Patent Application Serial No. 09/464,283, filed December 15, 1999 (the subject matter of which is incorporated herein by reference).

[0016] Figure 4 is a flow chart that illustrates a method in accordance with an embodiment of the invention. In step 300, data from host memory that is to make up a part of the payload of a TCP message is transferred from host memory 203 to DRAM 204 via PCI bus 202. Hardware in the path of this data determines a first checksum CSUM1 on the fly as the data passes by. This flow of data from host memory 203 to DRAM 204 is indicated on Figure 3 by arrow 205.

[0017] Although it could be in some situations, in the presently described example not all the data that will make up the TCP data payload is present in the same place in host memory 203. Consequently, the flow of data for the data payload from host memory 203 to DRAM 204 occurs in multiple different data moves as the various different pieces of the data are located and transferred to DRAM 204.

[0018] In step 301, more of the data that is to make the data payload of the TCP message is moved from host memory 203 to DRAM 204. A second checksum CSUM2 is generated as the data passes through the data path. This data flow is again represented by arrow 205.

[0019] In this example, the data payload is transferred to DRAM in three pieces. In step 302, the last of the data is moved from host memory 203 to DRAM 204 and a third checksum CSUM3 associated with this data is generated.

[0020] Processor 206, before this transferring is completed, builds in SRAM 207 the TCP header 208 that is to go on the TCP message. Processor 206 does not have all the data for the TCP payload so it cannot determine the complete checksum for the TCP message. It does, however, generate a checksum HDR CSUM 209 for the remainder (header portion 216) of the TCP header. This HDR CSUM is a partial checksum. Arrow 210 in Figure 3 illustrates the building

of the pseudoheader 208 (header portion 216 and partial checksum HDR CSUM 209) in SRAM 207.

[0021] In step 303, while the data payload is being transferred from host memory 203 to DRAM 204 in steps 301-302, the TCP header with the partial checksum HDR CSUM is moved from SRAM 207 to DRAM 204. This transfer is illustrated in Figure 3 by arrow 211.

[0022] In step 304, after all the data for the data payload has been transferred such that checksums for all the various pieces of the data payload have been generated, processor 206 combines those various data checksums together to form a single checksum for the data payload. In this example, there are three data checksums CSUM1, CSUM2 and CSUM 3. These are combined together to make a single data checksum DATA CSUM for the data payload. Processor 206 then supplies this DATA CSUM to a transmit sequencer 212. For additional details on one particular example of a transmit sequencer, see U.S. Patent Application Serial No. 09/464,283 (the subject matter of which is incorporated herein by reference). The supplying of the DATA CSUM to transmit sequencer 212 is illustrated in Figure 3 by arrow 213. At this point, the data payload is present in one place in DRAM 204 in assembled form with the pseudoheader 208 (header portion 216 and HDR CSUM 209) that was transferred from SRAM 207 to DRAM 204 in step 303. This assembly is a pseudopacket (pseudoheader and data payload). It is complete but for the fact that the header does not contain a complete checksum but rather contains the partial checksum HDR CSUM 209.

[0023] In step 305, the transmit sequencer 212 begins transferring the pseudopacket out of DRAM 204 for transmission onto a network 214. Network 214 is, in one

embodiment, a local area network (LAN). Transmit sequencer 212 combines the DATA CSUM with the HDR CSUM to create a final checksum and inserts the final checksum into the pseudopacket as the pseudopacket passes over path 215 from DRAM 204 to network 214. What is transferred onto network 214 is therefore a TCP packet having a correct TCP header with a correct checksum.

[0024] Although the functionality of the INIC is described here as being carried out on a separate card, it is to be understood that in some embodiments the functionality of the INIC is carried out on the host computer itself, for example on the motherboard of the host computer. Functionality of the INIC can be incorporated into the host such that payload data from host memory does not pass over a bus such as the PCI bus, but rather the INIC functionality is incorporated into the host in the form of an I/O integrated circuit chip or integrated circuit chip set that is coupled directly to the host memory bus. The I/O integrated circuit chip has a dedicated hardware interface for network communications. Where the INIC functionality is embodied in such an I/O integrated circuit chip, payload data from host memory is transferred to the network from the host memory by passing through the host's local bus, onto the I/O integrated circuit chip, and from the I/O integrated circuit chip's network interface port substantially directly to the network (through a physical layer interface device (PHY)) without passing over any expansion card bus.

[0025] Although the present invention has been described in connection with certain specific embodiments for instructional purposes, the present invention is not limited therefore. The present invention extends to packet

protocols other than the TCP protocol. In some embodiments, the first part of the packet is output from the INIC before the final checksum is inserted into the packet. The combining of the DATA CSUM and the HDR CSUM need not be performed by a sequencer and the pseudoheader need not be created by a processor. Other types of hardware and software can be employed to carry out these functions in certain embodiments. In some embodiments, the pseudoheader is assembled in memory or registers inside processor 109 rather than in a separate memory such as SRAM 111. Accordingly, various modifications, adaptations, and combinations of various features of the described embodiments can be practiced without departing from the scope of the invention as set forth in the claims.